

# Real-Time News Summarization with Adaptation to Media Attention

Andreas Rücklé<sup>†</sup> and Iryna Gurevych<sup>†‡</sup>

<sup>†</sup>Ubiquitous Knowledge Processing Lab (UKP)  
Department of Computer Science, Technische Universität Darmstadt

<sup>‡</sup>Ubiquitous Knowledge Processing Lab (UKP-DIPF)  
German Institute for Educational Research

[www.ukp.tu-darmstadt.de](http://www.ukp.tu-darmstadt.de)

## Abstract

Real-time summarization of news events (RTS) allows persons to stay up-to-date on important topics that develop over time. With the occurrence of major sub-events, media attention increases and a large number of news articles are published. We propose a summarization approach that detects such changes and selects a suitable summarization configuration at run-time. In particular, at times with high media attention, our approach exploits the redundancy in content to produce a more precise summary and avoid emitting redundant information. We find that our approach significantly outperforms a strong non-adaptive RTS baseline in terms of the emitted summary updates and achieves the best results on a recent web-scale dataset. It can successfully be applied to a different real-world dataset without requiring additional modifications.

## 1 Introduction

Important events such as natural disasters, protests, and accidents often trigger an increased information need for many people. These events usually develop over time with the occurrence of multiple sub-events, where publishers on the web create news articles on the topic while the situation is still developing. To stay fully updated, interested persons have to digest a substantial amount of information, which is not feasible in most cases. Some publishers therefore create real-time newsfeeds for selected high-impact events that are regularly updated with short texts to provide a live summary on the recent developments. An excerpt of an example summary is shown in Figure 1. Because the updates are usually created by journalists, the process is laborious and can only be applied to few events.

15/02/2013 - 08:12
A meteor shower caused several explosions in the lower atmosphere above Russia's Urals region.
15/02/2013 - 08:15
Multiple people were injured by broken glass.
15/02/2013 - 08:30
The emergency ministry issued a statement saying that no civil aircraft or local power stations were damaged.

Figure 1: Three updates for an example summary of the event *Russian Meteor* (2013).

Automatic approaches to real-time summarization (RTS) on the other hand can generate live summaries for a large number of events without entailing additional editorial cost (Aslam et al., 2014). This summarization process is different to retrospective approaches because all news articles must be processed in a timely fashion as soon as they are available. Thus, real-time in this context refers to the continuous decision-making process over an unbounded stream of news articles where each input document can trigger the emission of new updates.

To deal with this challenge, current approaches to RTS use real-time sentence filtering methods with different heuristics (McCreadie et al., 2014a; Raza et al., 2015) or more complex, real-time capable learning to search methods (Kedzie et al., 2016). They apply the same methods over the full timeframe of an event without explicit adaptation to important changes. However, when major sub-events occur, there is a sudden increase in media attention with a large number of news articles being published on the topic. We hypothesize that the detection of these changes to adapt the summarization process to media attention allows us to create an improved event summary.

In this work, we present an approach to RTS that adapts to changes in news events at run-time

by explicitly switching between configurations that determine important parameter choices for summarization. Within our approach, we combine simple yet effective methods for document filtering, single document summarization, and redundancy detection, which is inspired by previous work (McCreadie et al., 2014a). To adjust the parameters of these methods according to important changes in the news events, we continuously predict media attention by measuring moving averages of the number of relevant news articles over time. We switch the summarization configurations according to a ruleset whenever we detect significant changes in our predictions. This allows us to exploit redundancies in content at times with higher media attention to produce more precise updates for the summary of the news event.

Our two main contributions are as follows. First, we show that media attention is an important attribute that can be utilized for improving approaches to RTS. As a result, our approach is able to achieve the best results on a recent web-scale dataset, and can successfully be applied to a different real-world dataset without requiring additional modifications. Second, we demonstrate that simple methods for document filtering, single document summarization, and redundancy detection are very effective for RTS if suitably configured at run-time.

## 2 Related Work

RTS is strongly related to update summarization, where the goal is to create an update summary with only new and changed information based on a previous summary and a small set of new documents (Dang and Owczarzak, 2008). Early approaches apply standard multi document summarization methods followed by a redundancy removal step (Fisher and Roark, 2008; Copeck et al., 2008), whereas more recent approaches incorporate topic models (Delort and Alfonseca, 2012; Conroy et al., 2011) or specialized sentence re-ranking methods (Du et al., 2010; Li et al., 2013, 2015).

The periodical application of update summarization makes it possible to summarize long-running events that develop over a period of several weeks. McCreadie et al. (2014b), for example, use this approach and select sentences from hourly update summaries according to their prevalence and novelty. A major disadvantage, however, is the inability of being real-time capable. Similar areas are retrospective temporal summarization (Allan et al.,

2001) and on-line temporal summarization (Guo et al., 2013).

To accelerate research within summarization of long-running events, the TREC temporal summarization (TREC-TS) tracks were initiated (Aslam et al., 2014). The goal is the emission of updates at arbitrary times based on a large stream of input documents and an event query. Some approaches that use the TREC-TS datasets rely on incremental techniques to create updates over regular time windows. Kedzie et al. (2015), for example, use an incremental salience prediction method and a clustering approach to emit updates in hourly intervals. Other approaches are also real-time capable. McCreadie et al. (2014a) rely on simple filtering and redundancy detection methods and feature-based sentence extraction. Kedzie et al. (2016) use a real-time sequential decision-making process by adapting a learning to search approach. And Raza et al. (2015) rely on cosine-similarity heuristics to emit only the first sentence of relevant news articles.

## 3 Real-Time News Summarization (RTS)

**Problem Definition** Given a stream of input documents (i.e. news articles)  $S_{in} \leftarrow d_1, d_2, \dots, d_n, \dots$  and an event topic in the form of a query  $q$ , we want to emit a stream of output sentences  $S_{out} \leftarrow u_1, u_2, \dots, u_m, \dots$  with new and important information related to  $q$ . The output sentences are referred to as *updates* whereas the output stream itself is denoted as the *summary*. Each document  $d_i \in S_{in}$  is associated with a timestamp  $t_i$  where  $t_i \leq t_{i+1}$ . This reflects a real-life scenario where incoming documents are analyzed in the same order as they are published. Importantly, every document  $d_i$  invokes a decision-making process that can lead to the emission of new updates.

**Our Approach to RTS** We rely on a multi-step approach with three separate responsibilities: First, we filter  $S_{in}$  in regard to  $q$ . Second, we process the remaining relevant documents with a single document summarization method and extract the most important sentences. And third, for every extracted sentence, we decide if a new update should be emitted to  $S_{out}$ . See Figure 2 for a visualization.

Our approach is similar to the work of McCreadie et al. (2014a) who also rely on a processing pipeline. We however do not bind the individual steps to any particular algorithm. The benefit of this approach is the ability to re-configure all individual responsibilities separately at run-time.

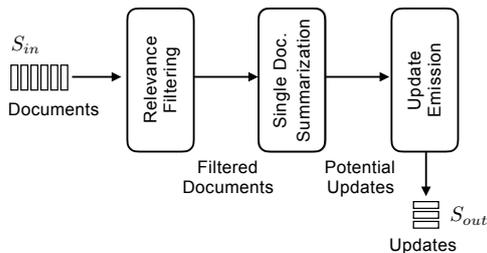


Figure 2: Our multi-step approach to RTS.

## 4 Adaptation to Media Attention

**Measurement of the News Stream** To explicitly adapt our approach to media attention at run-time, we continuously measure the stream of news articles in regard to the event query. We calculate moving averages for the number of news articles that pass the document filtering over time windows of 6 (*MA6*) and 24 hours (*MA24*). Moving averages enable us to suppress a certain amount of expected volatility while still being sensitive to important changes. *MA6* (over  $\frac{1}{4}$  day) and *MA24* (over a full day) thereby allow us to quickly react to increases in media attention (*MA6*) while ignoring common periodical changes, for example day vs. night (*MA24*). A visualization of the moving averages for two events is shown in Figure 4.

With these continuous measurements, we can detect increases in media attention by scanning for sudden increases in *MA6*. We can also detect decreases in media attention by observing decreasing values of both *MA6* and *MA24*. This enables our approach to select of a suitable configuration for summarization at run-time.

**Configuration Selection** Our approach can choose from a list of configurations  $\Psi$  at run-time, where each configuration  $\psi \in \Psi$  determines important parameters for relevance filtering, single document summarization, and update emission. Thus, the two most important properties are  $\Psi$  and the behavior to select configurations.

We perform the selection as follows. At the beginning of an event we always select the start configuration  $\psi_{current} = \psi_{start}$ . During summarization, we obtain important information about media attention of the event through continuous measurements of *MA6* and *MA24*. Based on this information together with  $\psi_{current}$ , our approach continuously evaluates a list of transition rules that define conditions for configuration switching. When a rule triggers a switch, the new configuration is im-

mediately selected and all related parameters are changed accordingly.

The transition rules together with the configurations  $\Psi$  and the continuous predictions of *MA6* and *MA24* enable our approach to explicitly adapt to the event at run-time. In the following, we describe the methods we use in the individual RTS steps and outline all relevant configuration parameters. We present the different configurations and transition rules later in Section 7.

## 5 Summarization Methods

**Document Filtering** We use a simple term-based filtering approach to determine the relevance of a news article  $d$  in regard to the event query  $q$ . If all stemmed words of  $q$  appear in the first  $n$  sentences of  $d$  and at least twice in the full text, we consider  $d$  as relevant. Otherwise we discard  $d$ .  $n$  is an important parameter that is determined by the selected configuration.

This approach to filtering is motivated by the inverted pyramid, which states that news stories usually begin with a story lead that contains the most important information followed by the article body with additional details (Pöttker, 2003).

**Single Document Summarization** We use the greedy summarization method MMR, which extracts summary sentences by minimizing the summary redundancy and maximizing the query similarity (Carbonell and Goldstein, 1998). The number of extracted sentences  $m$  is determined by the selected configuration.

We rely on cosine similarity with  $tf \cdot idf$  scores to measure the similarity of sentences in MMR.  $idf$  is approximated by the inverse term count over the static corpus web1t (unigrams).<sup>1</sup> We set the MMR balancing parameter to  $\lambda = 0.5$ , a common choice to not favor query similarity over redundancy detection. We only consider sentences for extraction that contain between 7 and 30 non-stop words and a named entity. Similar heuristics were applied by McCreadie et al. (2014a).

**Update Emission** Each individual sentence that was extracted in the prior step invokes a decision-making process for the update emission to determine if a new update  $u$  should be emitted to the summary  $S_{out}$ . Our approach follows the intuition

<sup>1</sup><https://catalog.ldc.upenn.edu/LDC2006T13>

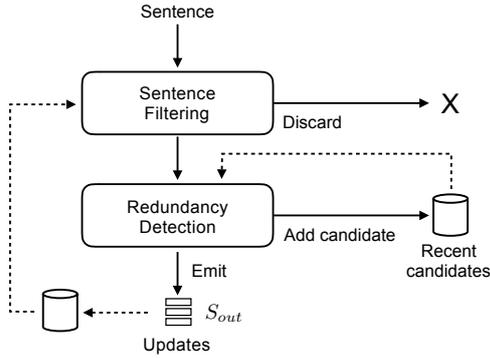


Figure 3: The update emission process.

that in case of important sub-events multiple publishers report about the incident at similar times. We assume that the exploitation of redundancies in content can help us to find important information. Because the amount of available redundant content is proportional to the media attention, run-time adaptation is required.

Our approach is visualized in Figure 3. First, we discard sentences that are redundant to previously emitted updates. We use the same cosine similarity scoring method as described previously and discard sentences if their similarity to a previous update exceeds the threshold  $t_s$ . We furthermore employ a Naïve Bayes classifier to discard (obviously) irrelevant sentences,<sup>2</sup> which we trained on manually annotated sentences from the TREC-TS 2013 dataset. We only rely on simple features like term count, frequency of uppercase letters, and frequency of non-alphanumeric letters.

After sentence filtering we apply a real-time capable redundancy detection method. For a sentence  $s$ , we check the similarity against a list of recently stored candidate sentences that were not emitted as updates. If we find at least  $g$  sentences with similarity greater than a threshold  $t_c$ , we emit one sentence from this group as an update. The emitted sentence is the one with the highest similarity to all other sentences of the group. Otherwise, if we cannot find enough similar sentences, we add  $s$  itself to the list of candidates.

Parameters that are set by the chosen configuration are  $t_s$ ,  $t_c$ , and  $g$ .  $g$  is especially important because it determines the required redundancy.

**Confidence Scoring** In our evaluation, which we describe in Section 6, we rely on manual judge-

<sup>2</sup>E.g. “CBS News CBSNews.com - CBS Evening News - CBS This Morning - 48 Hours”

Dataset	News Articles	News Articles per Hour	Avg. Event Duration
2014	6,488,989	2,267	310 [h]
2015	145,266	36	186 [h]

Table 1: Statistics of the employed datasets.

ments of the top- $n$  updates for each summary. Thus, it is important to calculate a confidence score that allows us to find these top- $n$  updates.

For an update  $u$ , we calculate three different quality indicators that are derived from the group  $G$  of redundant sentences that are found in the redundancy detection step of the update emission, and their timestamps  $T$ :

$$c_c = \frac{1}{|G \setminus \{u\}|} \sum_{u_{group} \in G \setminus \{u\}} sim(u, u_{group})$$

$$c_t = \frac{max(0, 24 - max(T) + min(T))}{24}$$

$$c_o = 1 + 0.2 \cdot |G|$$

where  $c_c$  is the coherence measured by the average similarity ( $sim$ ) of the redundant sentences,  $c_t$  is the timeliness measured by the distance between earliest and latest timestamp (normalized by 24h), and  $c_o$  is a value derived from the group size (required redundancy).<sup>3</sup> We calculate the final confidence score as the product of these three indicators.

## 6 Experimental Setup

**Datasets** For our experiments we use the TREC-TS corpora of 2014 and 2015. Both are filtered versions of the larger TREC-KBA corpus that contains 1.2 billion web documents (Frank et al., 2012). All documents are timestamped, which allows us to simulate an ordered input stream. We only use news articles and filter out social media content. Besides web documents, the corpora also contain events queries (e.g. *2013 Eastern Australia floods*) and textual *nuggets*, which describe important sub-events (e.g. *Moonie highway flooded*). Nuggets form the gold-standard of information that should be included in a good summary for an event. Dataset statistics are listed in Table 1. Most notable, the 2015 corpus contains significantly fewer news articles per hour and the event duration is 40% shorter on average. We perform experiments on both datasets to compare approaches within different scenarios.

<sup>3</sup>For each update in the group, we add a 0.2 increase for  $c_o$ , which is motivated by a theoretical group size limit of 5.

We split the 2014 corpus into 4 development events<sup>4</sup> and 11 test events. We use all 21 events of the 2015 corpus for testing.

**Metrics** We adopt the evaluation metrics of TREC-TS 2014, which allows us to score the summary precision, recall, and timeliness. The metrics are heavily dependent on matchings between summary updates and nuggets, where a nugget matches an update whenever the nugget information is contained in the update. For space reasons we refer the reader to (Aslam et al., 2014) for a formal definition of the metrics. We briefly outline them below.

- *nEG* (Normalized Expected Gain): Measures the expected gain per update ( $\sim$  expected relevancy of updates). This is approximated by the number of nuggets a typical update covers. For each nugget, only the first match is considered. This is a *precision* metric.
- *C* (Comprehensiveness): The ratio of nuggets that have matches (weighted by nugget importance). Measures the amount of relevant content included. This is a *recall* metric.
- *EL* (Expected Latency): Timeliness of update timestamps compared to nugget timestamps.<sup>5</sup> It measures how fast important information is emitted (larger values = better). This is a *latency* metric.
- *H*: Harmonic mean of a latency-discounted variant of *nEG* and *C*.

**Annotations for Evaluation** We conducted our own annotation studies in accordance to the official TREC-TS track evaluations to obtain matchings between summary updates and event nuggets for all evaluated approaches. We employed three annotators for every event/approach combination who each matched the top-60 updates (determined by confidence score) against the event nuggets. The employed annotators were students with a linguistics and computer science background and prior annotation experience. For the remaining updates (not in top-60) we used exact matches from the pool of past track evaluations. In our results we calculate the mean of the individual scores derived from each annotator.

We measure an inter-annotator agreement of  $\kappa = 0.40$  (Cohen’s Kappa) on the 2014 dataset and  $\kappa =$

<sup>4</sup>Development events: *Boston Marathon Bombing*, *Costa Concordia disaster and recovery*, *2012 Afghanistan Quran burning protests*, and *2013 Eastern Australia floods*.

<sup>5</sup>The update timestamp is set to the timestamp of the last processed document in  $S_{in}$ .

0.56 on the 2015 dataset (moderate agreement). Previous work with comparable annotation studies reports similar results (McCreadie et al., 2014b).

**Evaluated Approaches** We primarily evaluate two different approaches. First, we test our approach with adaptation to media attention (RTS-Adap). The list of configurations and the transition rules are described in Section 7. Second, we evaluate a non-adaptive variant (RTS-Baseline). Compared to RTS-Adap, it relies on the same filtering and single document summarization methods, but employs a reduced update emission step. RTS-Baseline only executes the sentence filtering and skips the redundancy detection. This allows us to choose static configuration parameters, which we determined on the development events. Resulting values are  $n = 5$  for the number of sentences that are considered as article lead in the document filtering,  $m = 2$  for the number of single document summary sentences, and  $t_s = 0.3$  for the similarity threshold to discard updates.

We additionally re-evaluated top-performing systems from the TREC-TS tracks to provide a better overall comparison. We obtained the summary updates from the respective authors.

## 7 Configurations and Transition Rules

For RTS-Adap, we determined three different configurations  $\psi_a$ ,  $\psi_b$  and  $\psi_c$  that are suitable to summarize each of the development events. These configurations were obtained on the development events (using manual annotations for the matchings). The final values for each configuration are shown in Table 2. Whereas  $\psi_a$  and  $\psi_b$  only differ in the number of required sentences for redundancy detection in the update emission,  $\psi_c$  uses a different redundancy threshold and an increased number of sentences that are extracted within single document summarization. Furthermore, the document filtering is less restrictive, where only one token needs to be present in the document text twice (instead of all tokens). This is necessary to handle events with particularly low media attention.

With the individual configurations and the evaluation results on the development events, we determined the list of transition rules. We formulated different constraints that were necessary to obtain a good summary based on the results of the previous parameter search and manually optimized the transition rules to fulfill as many of the constraints as possible. Results are listed in Table 3. A visual-

Parameter	$\psi_a$	$\psi_b$	$\psi_c$
$n$ (document filter: article lead)	5	5	20
$m$ (document sum.: extracted sents)	4	4	5
$g$ (emission: redundant candidates)	2	1	1
$t_s$ (emission: update threshold)	0.3	0.3	0.3
$t_c$ (emission: candidate threshold)	0.6	0.6	0.45

Table 2: Parameter values that are determined by the three different configurations.

$\psi_{current}$	Condition	Change to
$\psi_c$	$MA6 > 6$	$\psi_b$
$\psi_b$	$MA6 > 14$	$\psi_a$
$\psi_a$	$MA24 < 6$ and $MA6 < 6$	$\psi_b$
$\psi_b$	$MA24 < 1$ and $MA6 < 1$	$\psi_c$

Start configuration:  $\psi_c$

Table 3: Ruleset for configuration switching.

ization of the adaptive configuration selection with these rules for two events is shown in Figure 4.

## 8 Experimental Results

**2014 Dataset** In the first experiment, we study the question of the added value of RTS-Adap compared to RTS-Baseline. We also re-evaluated the best-performing approach of TREC-TS 2014 (CUNLP-AP), which is based on an affinity propagation clustering method (Kedzie et al., 2015).

The results are shown in Table 4. In particular, RTS-Adap outperforms our baseline on all metrics with a significant improvement on  $\mathcal{H}$ . Most notable, it substantially increases the strong results for the precision-oriented metric  $nEG$ . At the same time RTS-Adap also achieves significantly better latency results. These improvements are a result of the effective exploitation of redundancies in content according to media attention, which allows only the most important and timely information to be emitted. Sentences from retrospective reports or opinion texts are usually discarded due to missing redundancies across recent news articles. Information that is already included in the summary is also discarded due to strict filtering in the update emission. Thus, RTS-Adap is highly effective in avoiding emitting irrelevant content.

In comparison to CUNLP-AP, our approach with adaptation to media attention achieves significantly better results on the precision-oriented metric  $nEG$ . Even though CUNLP-AP achieves better recall, the summaries of RTS-Adap are more balanced. This is particularly reflected in the combined metric  $\mathcal{H}$  where RTS-Adap outperforms CUNLP-AP by a

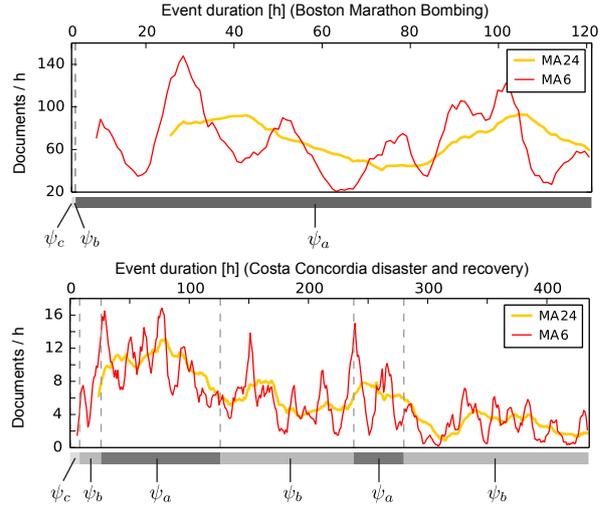


Figure 4: Configuration switches of RTS-Adap.

System	$C$	$nEG$	$EL$	$\mathcal{H}$
(c) CUNLP-AP	<b>0.32<sub>b</sub></b>	0.07	1.22	0.12
(b) Baseline	0.23	0.11 <sub>c</sub>	1.05	0.12
(a) RTS-Adap	0.26	<b>0.13<sub>c</sub></b>	<b>1.23<sub>b</sub></b>	<b>0.17<sub>b</sub></b>

Table 4: Results on 2014 data. Subscripts indicate statistical significance (Wilcoxon test,  $p < 0.05$ ).

substantial margin. The difference is not statistically significant due to a high variance in the result scores.<sup>6</sup> Additionally, compared to CUNLP-AP, our non-adaptive baseline achieves a similar result on  $\mathcal{H}$  because of high precision scores. This particularly demonstrates the effectiveness of the simple three-step approach to RTS.

In our second experiment, we study the performance of RTS-Adap compared to a static variant of the same approach that does not change configurations. To get a better impression of the adaptation itself, we evaluate the static approach for  $\psi_a$ ,  $\psi_b$ , and  $\psi_c$ . To keep annotation efforts at a feasible level, we selected five random events from our test set for this evaluation. Table 5 shows the results on  $\mathcal{H}$ . For only one event RTS-Adap does not select a suitable configuration. On the other hand, in three cases it achieves better results than the best possible individual configuration. This strongly suggests that our method is very effective because it can select the best possible configuration for individual event segments to create a better overall summary.

**2015 Dataset** In the third experiment, we study the question on the influence of a different dataset.

<sup>6</sup>Relative std. on  $\mathcal{H}$ : RTS-Adapt:  $\pm 47\%$ , CUNLP-AP  $\pm 73\%$ . RTS-Adapt produces more consistent results.

Event	$\psi_a$	$\psi_b$	$\psi_c$	RTS-Adap
Egyptian Riots	0.11	0.12	<b>0.15</b>	0.14
In Amenas	0.06	0.06	0.07	<b>0.11</b>
Russian Prot.	0.14	<b>0.17</b>	0.09	0.09
Russia Meteor	0.21	0.24	0.24	<b>0.29</b>
Southern Calif.	0.18	0.23	0.23	<b>0.25</b>

Table 5: A comparison of RTS-Adap against the same approach with static configurations ( $\mathcal{H}$ ).

System	$C$	$nEG$	$EL$	$\mathcal{H}$
(c) CUNLP-AP	0.27	0.06	1.04	0.07
(s) CUNLP-SD	<b>0.33</b>	0.11 <sub>c</sub>	<b>1.33</b> <sub>cb</sub>	0.18 <sub>c</sub>
(b) Baseline	0.32	0.10 <sub>c</sub>	1.23 <sub>c</sub>	0.15 <sub>c</sub>
(a) RTS-Adapt	0.32 <sub>cr</sub>	0.11 <sub>c</sub>	1.29 <sub>c</sub>	0.18 <sub>cb</sub>
(r) RTS-Adapt/Re	0.31	<b>0.11</b> <sub>csb</sub>	1.33 <sub>cb</sub>	<b>0.19</b> <sub>cb</sub>

Table 6: Results on 2015 data. Subscripts indicate statistical significance (Wilcoxon test,  $p < 0.05$ ).

Besides CUNLP-AP we also re-evaluated CUNLP-SD,<sup>7</sup> a top-performing approach of 2015 that is based on sequential decision-making with a learning to search method (Kedzie et al., 2016). We also created a version of RTS-Adap with conditions that were optimized on results from 2014 (RTS-Re).

Experimental results are listed in Table 6. Most notable, RTS-Adap can successfully be applied to a new real-world dataset without requiring a different ruleset. Compared to RTS-Baseline, our approach, again, achieves a better result on the combined metric  $\mathcal{H}$ , which is primarily due to better latency scores. On the other hand, the improvements in terms of precision and recall are much smaller. This is an effect of missing high-impact events in the dataset, which results in a small number of relevant news articles per event. This situation strongly favors simple approaches like RTS-Baseline that rely on simple content filtering. RTS-Adap however is still able to achieve better results compared to RTS-Baseline because it correctly selects configurations for low media attention. Our approach performs on the same level as CUNLP-SD and significantly outperforms CUNLP-AP on all measures, which especially shows the effectiveness of adaptation to media attention given its strong performance on the 2014 dataset. Even though RTS-Re achieves the best results in our evaluation, changes are relatively small. This suggests that our approach is robust against changes in the ruleset.

**Error Analysis** We identified two sources of errors within our approach. First, RTS-Adap some-

<sup>7</sup>Run id: 3LtoSfltr5

times selects the wrong configuration when an event is especially long-running with constant low media attention. An example is the event *Russian Protests* in Table 5. Here, our approach chooses the least restrictive configuration  $\psi_c$  for the full event timeframe, which results in multiple updates per day. Because the event is active for more than a month, the summary contains too much fine grained updates. As a solution, we could detect especially long-running events with the goal to select better suited configurations.

Second, our simple document filtering approach leads to misclassifications in some cases. As a result, irrelevant news articles are further processed and a small number of irrelevant updates are emitted. We can see this behavior in cases with misleading lexical overlap between the query and an unrelated input document. For example, a news article on Bulgaria protesting against an EU decision passes the filter for the unrelated event query *Bulgarian Protests* (against government). This problem could be solved by using more sophisticated document filtering methods.

## 9 Conclusion

In this work, we showed that media attention is an important attribute for RTS that can be utilized to improve event summaries. We presented an approach that automatically detects changes within media attention by continuously measuring moving averages for the number of relevant news articles over time. By switching summarization configurations at run-time, we can effectively exploit redundancies in content at times with high media attention and thereby create better, more precise summaries. Our experimental results showed the effectiveness of our approach, which significantly outperforms a strong non-adaptive baseline in terms of the emitted summary updates and achieves the best overall results on a recent web-scale dataset. Strong results on a different real-world dataset furthermore suggest that our approach can also be applied to other scenarios without requiring additional modifications in the employed ruleset. We showed that simple methods are highly effective within RTS if they are suitably configured at run-time.

## Acknowledgments

This work has been supported by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1).

## References

- James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. **Temporal Summaries of New Topics**. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New Orleans, LA, USA, pages 10–18. <https://doi.org/10.1145/383952.383954>.
- Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard Mccreadie, Virgil Pavlu, and Tet-suya Sakai. 2014. **TREC 2014 Temporal Summarization Track Overview**. *Proceedings of the 23rd Text REtrieval Conference (TREC 2014)*.
- Jaime G. Carbonell and Jade Goldstein. 1998. **The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries**. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Melbourne, Australia, pages 335–336. <https://doi.org/10.1145/290941.291025>.
- John M. Conroy, Judith D. Schlesinger, Jeff Kubina, Peter A. Rankel, and Dianne P. O’Leary. 2011. **CLASSY 2011 at TAC : Guided and Multi-lingual Summaries and Evaluation Metrics**. *Proceedings of the Text Analysis Conference*.
- Terry Copeck, Anna Kazantseva, Alistair Kennedy, Alex Kunadze, Diana Inkpen, and Stan Szpakowicz. 2008. **Update Summary Update**. *Proceedings of the Text Analysis Conference*.
- Hoa T. Dang and Karolina Owczarzak. 2008. **Overview of the TAC 2008 Update Summarization Task**. *Proceedings of the Text Analysis Conference*.
- Jean-Yves Delort and Enrique Alfonseca. 2012. **Dual-Sum: a Topic-Model based approach for update summarization**. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*,. Avignon, France, pages 214–223. <http://dl.acm.org/citation.cfm?id=2380845>.
- Pan Du, Jiafeng Guo, Jin Zhang, and Xueqi Cheng. 2010. **Manifold ranking with sink points for update summarization**. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. Toronto, ON, Canada, pages 1757–1760. <https://doi.org/10.1145/1871437.1871722>.
- Seeger Fisher and Brian Roark. 2008. **Query-focused supervised sentence ranking for update summaries**. *Proceedings of the Text Analysis Conference*.
- John R. Frank, Max Kleiman-Weiner, Daniel A. Roberts, Feng Niu, Ce Zhang, Christopher Ré, and Ian Soboroff. 2012. **Building an entity-centric stream filtering test collection for TREC 2012**. Technical report, DTIC Document.
- Qi Guo, Fernando Diaz, and Elad Yom-tov. 2013. **Updating Users about Time Critical Events**. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*. Moscow, Russia, pages 483–494. <http://dl.acm.org/citation.cfm?id=2458233>.
- Chris Kedzie, Fernando Diaz, and Kathleen McKeown. 2016. **Real-Time Web Scale Event Summarization Using Sequential Decision Making**. In *Proceedings of the International Joint Conference on Artificial Intelligence*. New York, NY, USA, pages 3754–3760. <http://www.ijcai.org/Abstract/16/528>.
- Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. **Predicting Salient Updates for Disaster Summarization**. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China, pages 1608–1617. <http://www.aclweb.org/anthology/P15-1155>.
- Chen Li, Yang Liu, and Lin Zhao. 2015. **Improving Update Summarization via Supervised ILP and Sentence Reranking**. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*. Denver, CO, USA, pages 1317–1322. <http://www.aclweb.org/anthology/N15-1145>.
- Xuan Li, Liang Du, and Yi D. Shen. 2013. **Update summarization via graph-based sentence ranking**. *IEEE Transactions on Knowledge and Data Engineering* 25(5):1162–1174. <https://doi.org/10.1109/TKDE.2012.42>.
- Richard McCreadie, Romain Deveaud, M-dyaa Albakour, Stuart Mackie, Nut Limsopatham, Craig Macdonald, Iadh Ounis, and Thibaut Thonet. 2014a. **University of Glasgow at TREC 2014 : Experiments with Terrier in Contextual Suggestion , Temporal Summarisation and Web Tracks**. *Proceedings of the 23rd Text REtrieval Conference (TREC 2014)*.
- Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014b. **Incremental Update Summarization: Adaptive Sentence Selection based on Prevalence and Novelty**. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. Shanghai, China, pages 301–310. <https://doi.org/10.1145/2661829.2661951>.
- Horst Pöttker. 2003. **News and its communicative quality: the inverted pyramid – when and why did it appear?** *Journalism Studies* 4(4):501–511.
- Ahsan Raza, Devin M. Rotondo, and Charles L. A. Clarke. 2015. **WaterlooClarke : TREC 2015 Temporal Summarization Track**. *Proceedings of the 24th Text REtrieval Conference (TREC 2015)*.